

# Team Eugeniusz

Seiji Yawata, Nobel Gautam, Farhan Haque, Gabriel Marks

## Overview

We are creating an online multiplayer rendition of the Tron Light Cycle Game. After you register and login, you can choose to create a new game, or join an existing game. The creator can select how many players can join the game, and decides which area they would like to play in. If you are the first one online, you are placed in a waiting queue until another player comes online. When another player comes online, you are placed into a game. Up to four players can play in the same game at the same time. In order to kill your opponent, you make them run into your trail. The last man standing gets a point. Whoever reaches 5 points first wins the game. Your stats for the game and your wins are recorded under your username.

## Style Guide

**Function+Variable Names-** camelCase

**Class Names-** CapsCamelCase

**Spacing-** soft (spaces) 4-space tabs

**Curly braces-** same line after a space, i.e.

```
foo = function() {  
    ...  
}
```

**Comments-** Same line: 2 spaces, comment marker, space, comment, i.e.

```
foo = 12 # foo must be 12  
var foo = 12; // 12 is essential  
Doc comments for all python functions:  
def a(b, c):  
    '''  
  
    a is a function that does stuff..  
    b -  
    c - '''  
    ...
```

## Components

### File Structure

```
eugeniusz/  
  app.py  
  templates/  
    login.html  
    game.html  
    user.html  
    server.html  
  data/  
    users.db  
  utils/  
    __init__.py  
    game.py  
    login.py  
    server.py  
  static/  
    game.js  
    wait.js
```

#### app.py

- **home()** - Routed to "/". If the user is logged in, run `server()`. Otherwise, `render_template login.html`.
- **server()** - Routed to "/server". If the user is not logged in, redirect them to "/". Otherwise, `render server.html`.
- **data()** - Routed to "/data". Used by JS files to get and receive matchmaking + game information from the Python backend.
- **user()** - Routed to "/user/<userid>". Renders `user.html` with relevant user stats and information from `users.db`.
- **game()** - "/game" route that will render `game.html` and allow the user to begin playing the game.

#### login.html

Presents a page with a form, which will allow users to login to begin using our site. If the user does not have an account, they are able to register on this page.

**game.html**

Runs game.js.

**user.html**

Takes score info about the user, and formats it into a nice page.

**server.html**

Server browser, allows you to create a server as well.

**users.db**

Will contain SQL databases which will store users with their passwords. Another database will store the user with all their scores.

**game.py**

```
class Player - info about one player relevant to the game
    int userid - user id
    float x - x position
    float y - y position
    int dir - direction facing (enumeration)
```

```
class Instance - One instance of a game
    int size - size of the playing field
    {int:Player} players - playernum:player for each
    player in game
    {int:int} scores - playernum:score
```

Instance newInstance(userid1, userid2) - makes a new game with the two users

Instance.update(userid, data) - takes data from a specific user and uses that to update game data

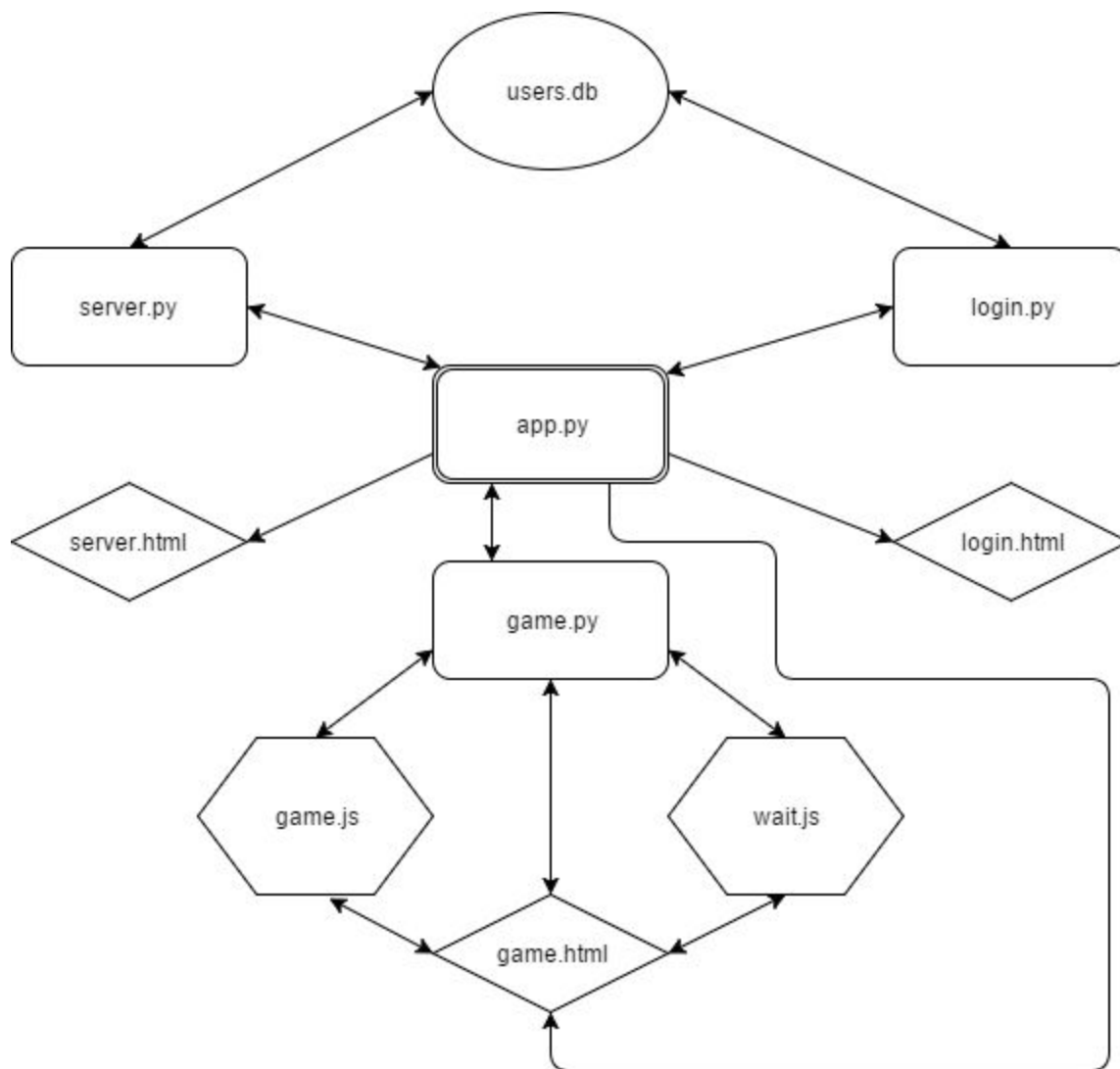
**wait.js**

Loops a request to /data that asks if this user has found a match. When the user finds a match, terminates.

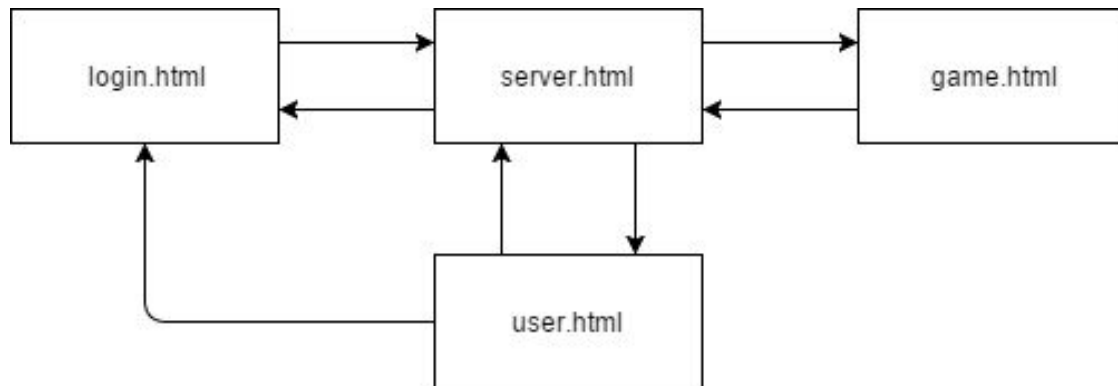
### **game.js**

Loops at a specific frame rate. At the beginning of each game loop, gets data from a socket to for the newest data for the current game, and sends current keypresses. Renders game objects based on data received.

### **Component Map**



## Site Map



The root of the website is server.html. It leads the user to the game itself in game.html. It also provides a way to log in/register (via login.html), and a link to user profile (via user.html).

## Database Schema

### Accounts

<b>TEXT</b> userid	<b>TEXT</b> name	<b>TEXT</b> password
User's assigned id	Username	Hashed password

### Scores

<b>TEXT</b> userid	<b>INT</b> myscore	<b>INT</b> otherscore
User's assigned id	User's score in game	Other player's score

## Division of Labor

Gabriel- Project Manager + Python backend

Nobel- Python backend

Seiji- Main JS backend

Farhan- Frontend + JS backend

## Schedule

Date	Task
5/15	Multiplayer test p1 (see if we can have two people online)
5/20	Multiplayer test p2 (see if we can have two people moving + seeing other characters move)
5/24	User database + working login
5/29	Graphics stuff
6/2	Walls + Death handling
6/6	Game ending
6/11	Server browser